

Makalah Nomor: KNSI-472

## PENCARIAN CITRA BERDASARKAN BENTUK DASAR TEPI OBJEK DAN KONTEN HISTOGRAM WARNA LOKAL

Barep Wicaksono<sup>1</sup>, Suryarini Widodo<sup>2</sup><sup>1,2</sup> Teknik Informatika, Universitas Gunadarma<sup>1,2</sup> Jl. Margonda Raya No. 100 Pondok Cina Depok<sup>1</sup> [bharep@student.gunadarma.ac.id](mailto:bharep@student.gunadarma.ac.id), <sup>2</sup> [srini@staff.gunadarma.ac.id](mailto:srini@staff.gunadarma.ac.id)

---

### Abstrak

*Content-Based Image Retrieval (CBIR)*, adalah suatu aplikasi *computer vision* yang digunakan untuk melakukan pencarian gambar-gambar digital pada suatu basis data. Penelitian ini mengaplikasikan konsep *Content Based Image Retrival* dengan mengembangkan program pencarian citra dengan mengekstraksi bentuk dasar tepi objek menggunakan filter *Canny*, menentukan kode rantai *Freeman* dari hasil deteksi tepi dan mendefinisikan bentuk dasar tepi objeknya. Setelah itu dilakukan ekstraksi fitur citra berdasarkan konten histogram warna lokal. Tahapan yang dilakukan pada pengekstraksian nilai konten warna adalah membagi citra menjadi 16 segmen, dan mengkuantisasi nilai sebaran warna dan menyimpannya di masing-masing histogram warna lokal setiap bagian citra. Alat ukur yang digunakan untuk memperoleh nilai hasil perbandingan kedua citra adalah *Euclidean distance*. Proses pencarian berdasarkan konten bentuk tepi objek dan histogram warna lokal menghasilkan daftar sepuluh citra dari yang paling mirip dan menghasilkan hasil pencarian citra dengan tingkat kemiripan 100%. Data yang digunakan pada penelitian ini 20 sampel citra bendera negara-negara di dunia yang berformat JPG atau PNG.

**Kata kunci :** *CBIR, histogram warna lokal, bentuk dasar tepi objek, filter canny, euclidean distance*

---

### 1. Pendahuluan [*Times New Roman 10, bold*]

Perkembangan teknologi informasi telah banyak melahirkan sistem dan aplikasi yang sangat bermanfaat dan membantu penggunaannya dalam menyelesaikan pekerjaan. Sistem informasi berbasis computer membutuhkan basis data untuk menyimpan setiap data dan informasi yang diperlukannya. Basis data tidak hanya digunakan untuk menyimpan informasi berupa teks atau angka namun basis data telah mampu menyimpan informasi dalam bentuk citra. Seiring dengan terus berkembangnya ukuran dari basis data citra, metode tradisional yang biasa digunakan dalam pencarian citra sudah tidak mungkin lagi digunakan. misalnya pencarian citra dengan kata kunci, kadang kala kata kunci (teks) yang dimasukkan tidak sesuai dengan citra yang diharapkan sehingga dengan memberikan kata kunci saja adalah tidak cukup, harus dikembangkan dengan metode lain untuk mengekstrak informasi citra yang dapat digunakan sebagai pengganti atau ditambahkan pada sistem kata kunci. Untuk itu dikembangkan suatu metode baru yaitu CBIR ( *Content Based Image Retrieval* ). *CBIR* adalah suatu aplikasi *computer vision* yang digunakan untuk melakukan pencarian gambar-

gambar digital pada suatu basis data, yang dianalisa dalam proses pencarian itu adalah *actual contents* (kandungan aktual) sebuah gambar. Istilah *content* pada konteks ini merujuk pada warna, bentuk, tekstur, atau informasi lain yang didapatkan dari gambar tersebut.

Algoritma yang dikembangkan Febriani menunjukkan bahwa pencarian menggunakan informasi kemiripan bentuk tepi objek dapat memberikan hasil yang kurang sempurna pada objek yang bentuknya sama tetapi memiliki warna yang berbeda [4].

Pencarian citra menggunakan informasi warna dengan histogram warna global (GCH), mengembalikan hasil yang tidak sesuai dengan persepsi visual. CGH hanya cocok jika pengguna mencari citra dalam sistem basis data yang hanya memperhatikan distribusi warna global suatu citra.

Tujuan dari penelitian ini adalah mengaplikasikan CBIR yang mampu menemukan gambar (yang dijadikan *query* dalam pencarian) yang memiliki persentase kemiripan yang mendekati dalam basis data gambar yang tersedia berbasis histogram warna dan representasi bentuk objek pada gambar. Citra *query* dan citra basis data yang

digunakan berupa citra dengan format JPG atau PNG.

Tahapan pencarian yang dilakukan adalah mengkonversi format file input citra menjadi JPG, kemudian dilakukan ekstraksi bentuk menggunakan *canny edge detection*, *freeman chain code* dan pendefinisian tepi objek, setelah di dapat nilai ekstraksi bentuk, maka dari hasil nilai tersebut dilakukan ekstraksi berdasarkan warna. Kemudian untuk menghitung tingkat kemiripan pencarian dilakukan penghitungan kemiripan menggunakan *euclidean distance*.

## 2. Landasan Teori

### 2.1. Content-Based Image Retrieval (CBIR)

Proses umum dari CBIR adalah pada citra yang menjadi *query* dilakukan proses ekstraksi fitur (*image contents*), begitu halnya dengan citra yang ada pada sekumpulan citra juga dilakukan proses seperti pada citra *query*. Parameter fitur citra yang dapat digunakan untuk retrieval pada sistem ini dapat berupa histogram, susunan warna, tekstur dan *shape* (bentuk), tipe spesifik dari obyek, tipe *event* tertentu, nama, individu, lokasi, emosi.

Dalam sistem tersebut, konten visual dari kumpulan citra dalam basis data citra diekstraksi dan dideskripsikan dalam bentuk vektor fitur multi dimensi. Konten ini disimpan dalam basis data konten. Untuk mencari citra dalam basis data, penggunaanya memerlukan citra masukkan (*query*), citra *query* ini kemudian di ekstraksi konten visualnya dan direpresentasikan dalam bentuk vektor konten. Kemiripan atau jarak antara vektor konten dari citra ruang dan citra *query* dihitung oleh proses pengurutan. Proses pengurutan diperlukan untuk melakukan proses pencarian yang cepat dan efisien. Umpan balik dari pengguna merupakan modifikasi dari proses pencarian citra untuk menghasilkan pencarian citra yang lebih presisi.

### 2.2. Histogram Warna Lokal

*Color histogram* adalah representasi distribusi warna dalam sebuah citra yang didapatkan dengan menghitung jumlah piksel dari setiap bagian range warna, secara tipikal dalam dua dimensi atau tiga dimensi. Histogram warna sangat efektif mengkarakterisasikan distribusi global dari warna dalam sebuah *image*.

Sedangkan histogram warna lokal (LCH) membagi gambar menjadi beberapa bagian dan kemudian mengambil histogram warna tiap bagian tadi. LCH memang berisi lebih banyak informasi tentang gambar, namun metode ini membutuhkan lebih banyak proses komputasi. Dalam ekstraksi menggunakan histogram warna lokal, langkah-langkahnya adalah sebagai berikut :

1. Segmentasi gambar ke dalam blok/bagian/segmen dan mendapatkan histogram warna lokal untuk setiap blok/segmen.
2. Membandingkan blok di lokasi yang sama dari dua gambar (jarak antara dua blok adalah jarak antara histogram warna mereka).
3. Penjumlahan jarak dari semua blok.

### 2.3. Bentuk Dasar Tepi Objek

Tiga jenis bentuk dasar objek yang digunakan dalam penelitian ini adalah :

- a. *Tepi dengan garis vertical, horizontal dan diagonal*.

Secara umum formulasi untuk garis lurus dapat dihitung dengan menggunakan persamaan linier, yang disajikan pada persamaan (1):

$$\begin{aligned} y &= ax + b & \text{atau} \\ x &= ay + b, & \text{dengan } b = 0 \end{aligned} \quad (1)$$

dimana  $\alpha$  = sudut penyimpangan. Untuk mempermudah dalam menentukan garis dan busur digunakan asumsi dan pendekatan-pendekatan sebagai berikut ini:

- a. garis *horizontal*, jika  $x = ay + b$  dengan  $x = 0$ .
- b. garis *vertical*, jika  $y = ax + b$ , dengan  $y = 0$ .
- c. garis *diagonal*, jika  $\alpha = 1$  terhadap sumbu x atau sumbu y.

- b. *Tepi dengan sudut*

Suatu objek dapat mengandung berbagai sudut, sedangkan sudut terbentuk dari dua buah tepi yang ujungnya saling berhimpit. Berdasarkan pada kondisi tersebut, maka jenis sudut yang terbentuk dapat menghasilkan beberapa jenis, antara lain siku-siku, sudut lancip, dan sudut tumpul. Untuk sudut siku-siku terbentuk bila kedua buah tepi saling tegak lurus, sudut lancip bila kedua tepi membentuk sudut kurang dari  $90^\circ$ , sedangkan sudut tumpul terbentuk bila kedua tepi berhimpitan membentuk sudut lebih besar dari  $90^\circ$ .

- c. *Tepi dengan jenis busur*

Selain membentuk sudut, gabungan dua tepi juga dapat membentuk busur yang direpresentasikan.

### 2.4. Euclidean Distance

Jarak *Euclidean* dapat dianggap sebagai jarak yang paling pendek antar dua titik. Jika sebagian dari suatu atribut obyek diukur dengan skala berbeda, maka ketika menggunakan fungsi jarak *Euclidean*, atribut dengan skala yang lebih besar boleh meliputi atribut yang terukur pada skala yang lebih kecil.

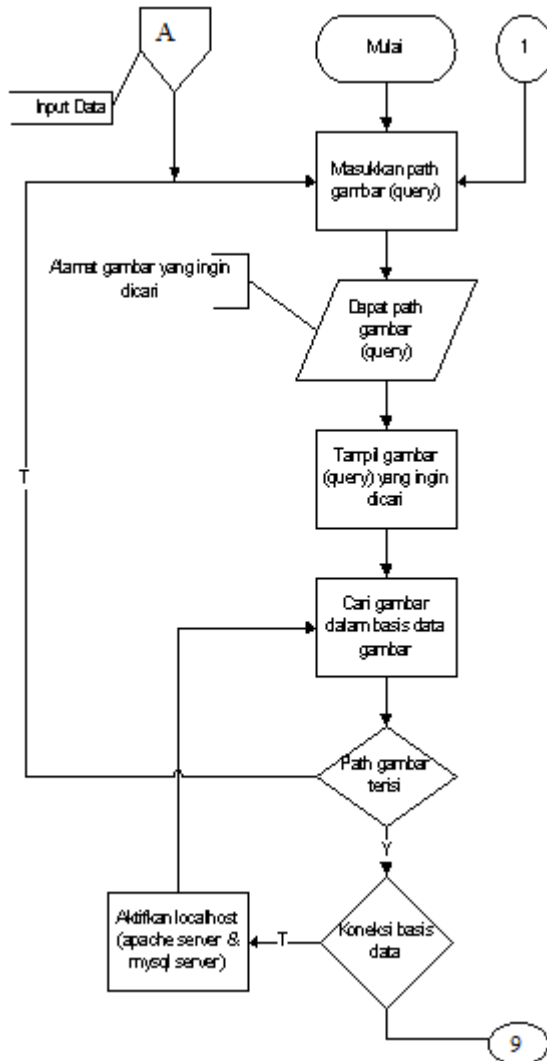
Titik antara p dan q merupakan panjang dari segmen pq. Rumus :

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2)$$

*Euclidean* sering digunakan karena penghitungan jarak dalam *distance space* ini merupakan jarak terpendek yang bisa didapatkan antara dua titik yang diperhitungkan

### 3. Pencarian Citra Berdasarkan LCH dan Bentuk Tepi Objek

Tahapan yang dilakukan dalam pembuatan aplikasi pencarian citra berdasarkan bentuk tepi objek dan histogram warna local adalah sebagai berikut :



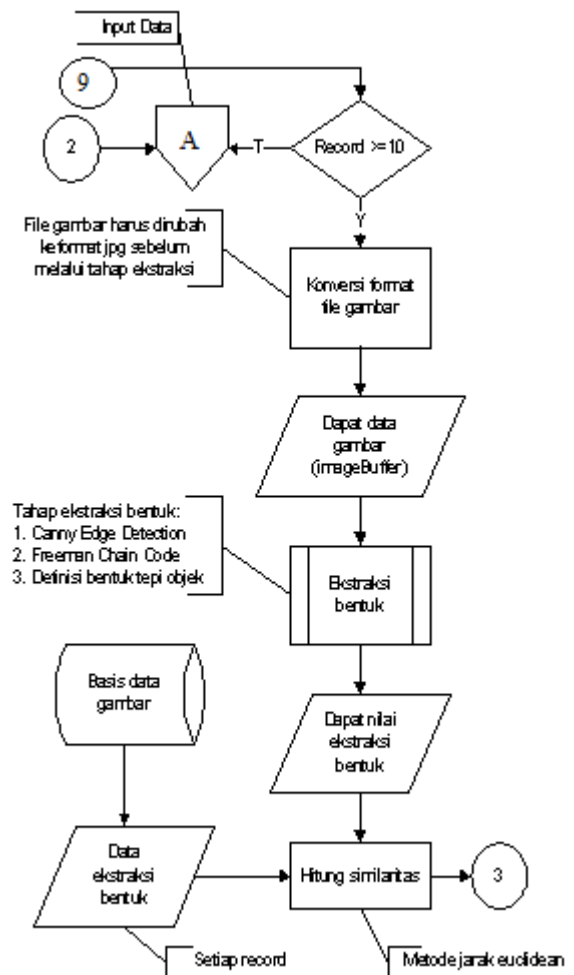
Gambar 1. Tahapan input citra aplikasi

Saat proses pencarian citra dijalankan, aplikasi terlebih dahulu melakukan proses validasi dengan mengecek kolom *path* citra sudah terisi, *basis data* sudah terkoneksi dan *record* basis data sudah terpenuhi. Jika salah satu parameter tersebut belum terpenuhi maka pengguna diminta untuk melengkapinya, misalnya jika lokal server belum dinyalakan (karena basis data menggunakan lokal server) maka akan keluar pesan untuk menyalakan lokal sever dan jika *record* dalam basis data masih kurang dari 10 (sepuluh) maka program akan

langsung membuka halaman *input* data agar pengguna dapat melengkapi batas minimal *record* yang diperlukan.



Gambar 2. Contoh citra input

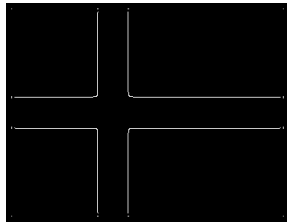


Gambar 3. Tahapan konversi format gambar dan ekstraksi bentuk

Setelah melalui tahap validasi, citra dikonversi terlebih dahulu ke dalam format jpg agar dapat diproses oleh kedua mesin ekstraksi. Hasil

konversi berupa *imageBuffer* yang digunakan sebagai masukan mesin ekstraksi.

Ekstraksi pertama yang diproses adalah ekstraksi bentuk dimana citra melalui beberapa proses *image processing* mulai dari deteksi tepi dengan menggunakan *filter Canny* kemudian hasil deteksi tepi tersebut ditelusuri untuk mendapatkan kode rantai tepi objek sehingga diperoleh data hasil ekstraksi bentuk tepi yang berupa himpunan bentuk tepi dasar objek yang diperoleh dari citra.



Gambar 4. Citra *Query* hasil deteksi tepi Canny




















Setelah itu citra melalui proses penelusuran tepi objek menggunakan metode freeman chain code.

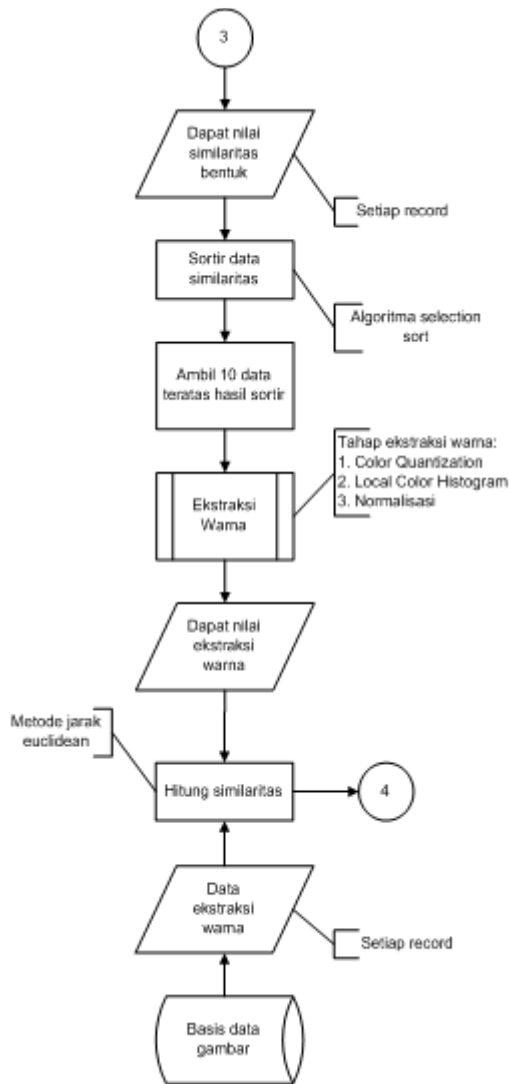
Tabel 1. Hasil proses kode rantai tepi citra

[illegible]

Untuk lebih jelasnya hal tersebut akan dijelaskan di bagian algoritma ekstraksi bentuk. Nilai ekstraksi bentuk yang telah diperoleh tersebut akan dibandingkan dengan semua *record* citra yang terdapat di basis data citra untuk memperoleh nilai *similaritas* (kemiripan) setiap citra dalam basis data terhadap citra *query*.

Tabel 2. Daftar *similaritas* bentuk hasil pengurutan

GQ - Gn	ID	Citra	Similaritas Bentuk
GQ - G1	11		0.0
GQ - G4	14		11.6562
GQ - G11	21		44.5352
GQ - G20	30		49.7749
GQ - G3	13		50.1237
GQ - G13	23		53.5621
GQ - G7	17		55.2444
GQ - G12	22		56.6053
GQ - G8	18		56.9319
GQ - G15	25		56.9319
GQ - G6	16		57.8609
GQ - G2	12		66.7785
GQ - G17	27		74.2061
GQ - G16	26		74.2128
GQ - G19	29		78.4359
GQ - G5	15		84.0476
GQ - G18	28		89.0616
GQ - G14	24		96.7039
GQ - G9	19		106.9885
GQ - G10	20		111.3424



Gambar 5. Tahapan ekstraksi warna

Data *similaritas* bentuk diurutkan dari nilai terkecil (serupa) hingga nilai terbesar (berbeda) dengan menggunakan algoritma pengurutan *selection sort*. Diambil sepuluh data teratas hasil pengurutan *similaritas* bentuk.

Ekstraksi kedua adalah ekstraksi warna yang menggunakan metode histogram warna lokal. Pada proses ini citra akan disegmentasi menjadi beberapa bagian, sehingga diperoleh nilai histogram warna setiap bagiannya. Untuk lebih jelasnya hal ini akan dijelaskan dibagian algoritma ekstraksi warna.

Tahap ekstraksi warna didahului dengan tahap segmentasi. Hal ini untuk membagi citra menjadi beberapa bagian dan memperoleh histogram warna lokal setiap bagiannya.



Gambar 6. Segmentasi citra *query*

Citra dimasukkan dengan format warna RGB diekstraksi untuk setiap piksel dari citra tersebut. Setiap piksel dalam *file* citra adalah array bertipe data integer dengan volume 4 bytes (32 bits). Sebagai contoh misal sebuah piksel memiliki nilai 11111111001100110011111000011110. Maka nilai piksel tersebut dapat dijabarkan:

- Byte pertama dari kiri [11111111] = Alpha (nilai transparansi citra).
- Byte kedua dari kiri [00110011] = Merah.
- Byte ketiga dari kiri [00111110] = Hijau.
- Byte keempat dari kiri [00011110] = Biru.

Untuk proses penelitian kali ini hanya memproses nilai warna Merah, Hijau dan Biru, karena citra masukan bertipe *file* jpg sehingga nilai alpha tidak akan mempengaruhi untuk proses selanjutnya.

Setelah itu dilakukan normalisasi untuk memperoleh hasil persepsi visual yang akurat terhadap dua citra yang memiliki distribusi warna sama walau memiliki ukuran yang berbeda ukuran. Nilai normalisasi histogram warna lokal citra diperoleh dengan merubah setiap format nilai sebaran warna menjadi nilai persentase sebaran warna untuk setiap bagian.











Tabel 3. Nilai normalisasi histogram lokal setiap bagian

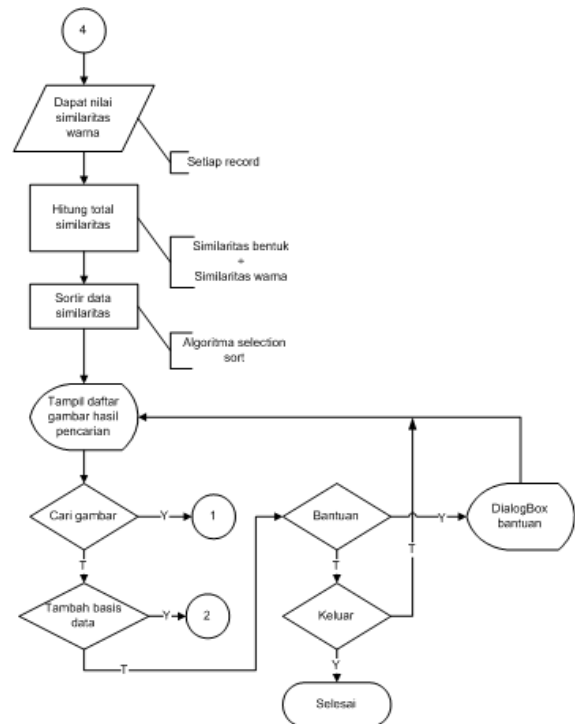
[illegible]

[illegible]

Dari proses ekstraksi ini diperoleh data nilai ekstraksi warna citra (*query*). Nilai ekstraksi warna yang telah diperoleh tersebut akan dibandingkan dengan sepuluh data hasil sortir *similaritas* bentuk sebelumnya untuk memperoleh nilai *similaritas* (kemiripan) berdasarkan warna untuk sepuluh data citra terhadap citra *query*.

Tabel 4. Daftar nilai *similaritas* warna

GQ - Gn	ID	Citra	Similaritas Warna
GQ – G1	11		0.0
GQ – G3	13		66.8162
GQ – G4	14		109.2211
GQ – G7	17		105.7193
GQ – G8	18		69.32440
GQ – G11	21		44.2422
GQ – G12	22		70.30923
GQ – G13	23		68.5539
GQ – G15	25		65.1897
GQ – G20	30		52.3542













Gambar 4. Tahapan hasil pencarian informasi

Setelah melalui tahap perbandingan berdasarkan bentuk dan warna, kemudian kedua nilai *similaritas* bentuk dan warna tersebut ditambah untuk setiap data. Nilai *similaritas* gabungan tersebut disortir kembali menggunakan algoritma pengurutan *selection sort*. Sehingga didapat daftar citra pencarian yang dimaksud.

Daftar citra pencarian tersebut berupa tabel dimana citra yang paling mirip dengan citra (*query*) akan berada di urutan paling atas. Daftar tersebut juga menampilkan nama deskripsi dan nilai *similaritas* dari hasil ekstraksi bentuk dan warna dan nilai *similaritas* gabungannya.

Nilai *similaritas* total dari setiap citra basis data kemudian diurutkan menggunakan metode selection sort. Hasil pengurutan dari nilai *similaritas* total ini yang ditampilkan di halaman hasil pencarian dalam bentuk daftar.

Tabel 5. Daftar *similaritas* total

GQ - Gn	ID	Citra	Similaritas Total
GQ – G1	11		0.0
GQ – G11	21		88.77749649528144
GQ – G20	30		102.12917971422931
GQ – G3	13		116.94000804609436
GQ – G4	14		120.87741370018833
GQ – G13	23		122.11604158454924
GQ – G15	25		122.12170853162134
GQ – G8	18		126.25640029271744
GQ – G12	22		126.914585544418
GQ – G7	17		160.96380870249638

#### 4. Kesimpulan

Berdasarkan hasil uji coba dan analisis hasil yang telah dilakukan menunjukkan bahwa pencarian citra berdasarkan bentuk dasar tepi objek dan konten histogram warna lokal mampu menemukan citra yang sama atau menyerupai dengan citra yang dicari. Dengan demikian dapat disimpulkan bahwa :

- Dari pengujian yang dilakukan dengan menggunakan 20 data citra bendera, didapatkan hasil pencarian 100 %.
- Alat ukur jarak yang digunakan untuk menentukan kemiripan citra adalah *Euclidean Distance* berhasil menemukan citra yang dicari dalam basis data dan mampu memberikan hasil pengidentifikasian citra yang serupa dengan citra *query*nya.

#### Daftar Pustaka:

- [1] Agus Sumarna, 2010, *CBIR Berdasarkan Ekstraksi Fitur Warna Menggunakan Java*, Jakarta, Universitas Gunadarma.
- [2] Anonim, 2012, *Ciri Berdasarkan Kode Rantai Chain Code*,  
<http://soemberilmu.files.wordpress.com/2009/01/t1koderantai.doc>.
- [3] Anonim, 2012, *Selection Sort Algoritma*.  
[http://www.algolist.net/Algorithms/Sorting/Selection\\_sort](http://www.algolist.net/Algorithms/Sorting/Selection_sort).
- [4] Febriani. 2010. *Pencarian Citra Berdasarkan Konten Bentuk Dasar Tepi Objek*, Jakarta, Universitas Gunadarma
- [5] Gibara, Tom, 2012, *Canny Edge Detection*.  
<http://www.tomgibara.com/computer-vision/canny-edge-detector>.
- [6] Kalra, Prem, 2007, *Canny Edge Detection*, New Delhi, Department of Computer Science & Engineering Indian Institute of Technology.